From Voxel to Intrinsic Surface Features

Olivier Monga, Peter Sander, Nicholas Ayache

From voxel to intrinsic surface features

Olivier Monga*, Nicholas Ayache* and Peter T Sander†

We establish a theoretical link between 3D edge detection and local surface approximation using uncertainty. As a practical application of the theory, we present a method for computing typical curvature features from 3D medical images. We determine the uncertainties inherent in edge (and surface) detection in 2- and 3-dimensional images by quantitatively analysing the uncertainty in edge position, orientation and magnitude produced by the multidimensional (2D and 3D) versions of the Deriche–Canny recursive separable edge-detector. The uncertainty is shown to depend on edge orientation, e.g. the position uncertainty may vary with a ratio larger than 2.8 in the 2D case, and 3.5 in the 3D case. These uncertainties are then used to compute local geometric models (quadric surface patches) of the surface, which are suitable for reliably estimating local surface characteristics; for example, Gaussian and Mean curvature. We demonstrate the effectiveness of our methods compared to previous techniques. These curvatures are then used to obtain more structured features such as curvature extrema and lines of curvature extrema. The final goal is to extract robust geometric features on which registration and/or tracking procedures can rely.

Keywords: typical surface features, local curvature extrema, mean and Gaussian curvature, local surface modelling, uncertainty, 3D edge detection

Modern medical imaging techniques, such as Magnetic Resonance Imaging (MRI) or X-ray computed tomography, provide three dimensional (3D) images of internal structures of the body, usually by means of a stack of tomographic images. In many applications, the physician asks for a segmentation of these 3D images into regions of interest he wants to manipulate, display, and characterize by objective measurements[1]. The first stage in the automatic analysis of such data is 3D edge detection[2–4] which provide points corresponding to the boundaries of the surfaces forming the 3D structure. The next stage is to characterize the local geometry of these surfaces to extract points or lines on which registration and/or tracking procedures can rely[5–8].

Sander and Zucker[9,10] have proposed to compute surface singularities by the calculation of curvatures using local approximation and by iterative refinement of the curvature field. In this paper we present a pipeline of processes which define a hierarchical description of the second order differential characteristics of the surfaces. We focus on the theoretical coherence of these levels of representation. Our levels of representation of the local geometry of the surfaces are:

- 3D edge points.
- Mean and Gaussian curvature, principal curvature directions.
- Local images of the curvatures.
- Characteristic points: curvature extrema, parabolic points, umbilic points.
- Characteristic lines: lines of curvature extrema, parabolic lines, umbilic points.

Three-dimensional edge detection is performed using recursive separable filters approximating the gradient or Laplacian, as described elsewhere[3,4,11]. From these edge points we build an adjacency graph with position and gradient vector attached to each edge point.

To compute curvatures from this graph we fit a local model at the neighbourhood of each point. The local model is a quadratic surface and the fitting method is a Kalman filter. Our approximation scheme uses the locations of the edge points and also the gradient direction which approximates the normal to the surface. Using uncertainty we establish a theoretical link between the edge detection and the local surface approximation by addressing the following two issues:

- How to determine the uncertainties inherent to edge detection in 2D or 3D images.
- How to incorporate these uncertainties into the computation of local geometric models.

In particular, we calculate the uncertainty of edge location, direction and magnitude for the 3D Deriche

*INRIA Domaine de Voluceau-Rocquencourt – B.P. 105, 78153 Le Chesnay Cedex, France
†INRIA, Unité de Recherche Sophia-Antipolis, 06565 Valbonne Cedex, France

Paper received: 7 February 1992

0262–8856/92/006403–15

operator. Our statistical results are then used as a solid theoretical foundation on which to base subsequent computations such as, for example, the determination of local surface curvature using local geometric models or for surface segmentation[12].

We tie the results of the analysis of the uncertainties involved in edge detection to the estimation of local geometric surfaces by a Kalman filtering technique. While the instantiation of these local models is similar to the method of Sander and Zucker[10], the utilization of uncertainties yields improved results. In addition, Kalman filtering permits incremental and selective incorporation of new data, thus ensuring that the local models are fit to, and only to, relevant data points. We expect that this will permit us to effectively deal with the problem of discontinuities where the local surface smoothness assumptions break down.

From the local fitting, we calculate for each edge point a mean curvature, a Gaussian curvature, and principal curvature directions, and covariance matrices defining the uncertainty. We define local curvature images by projecting at each point the curvatures of the neighbours onto the tangent plane. This yields the local aspect of the curvatures at a point.

From the curvature field we extract typical points such as curvature extrema, parabolic points, umbilic points. This is done by using the local image of the curvatures defined at each (edge) point. For instance, to select the extrema of the maximum curvature in the maximum curvature direction, we employ an algorithm very similar to the extraction of the extrema of the gradient magnitude in the gradient direction used in many 2D edge detection algorithms[13,14].

From these typical points we extract characteristic lines such as line of maximum curvature extrema and parabolic lines. To obtain lines of maximum curvature extrema, we perform a 3D hysteresis thresholding on the extrema curvature points using the maximum curvature. Here again the algorithm is very similar to that used to threshold the local gradient extrema described by Monga *et al.*[4].

The paper is organized as follows. In the next section we show how uncertainty allows unifying the 3D edge detection and the surface fitting. We deal first with the general problem of uncertainty in surface fitting, and then with the actual determination of the uncertainty from our particular operator (Deriche operator). We introduce the local geometric models and curvature computations, and set up the local surface fitting problem and the Kalman filter formalism for coping with uncertain data. The inherent uncertainty resulting from the 3D gradient operator is determined, along with how this is related to the Kalman filter.

In the third section we deal with the use of these curvatures to extract typical curvature features. We utilize the projection on the tangent plane of local images of curvatures which is frame invariant. Thus we reduce the extraction of typical curvature features to the filtering of the local images of curvatures.

The final section presents some experimental results for 3D synthetic images and real data. For real data we demonstrate the stability of the mean and Gaussian curvatures using two 3D scanner images of the same organ taken at different positions. We also show some results on the extraction of local curvature extrema on a MRI image of the face.

PARAMETRIC LOCAL SURFACE MODEL

Problem formulation

In this section we set up the local parametric surface models, describe briefly how to compute surface curvatures from the models, and present the problem of determining the parameters of the models from the data derived from the 3D gradient operator. We thus assume that we are given the locations of estimated surface points P and their estimated surface normals $\mathbf{N}$ corresponding to the vector gradient (see Monga and Deriche[3] and Monga *et al.*[4] on how to compute them). Using $\mathbf{N}$ we can establish a *tangent plane coordinate system* at P, which we denote $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$. Note that the basis $(\mathbf{P}, \mathbf{Q})$ of the tangent plane at P is arbitrary – the only constraint is that the coordinate system be right-handed and orthonormal.

In the following, P is the point at which the surface patch is being fit, and Q_i are neighbouring estimated surface points with associated normals $\mathbf{n}_i$, with both given in P's tangent plane coordinates (the development is simpler in these coordinates; we map everything into the actual image coordinates given below).

Local geometric model

We assume that the data returned by the gradient operator represent noisy estimates of points and normals from a (smooth) surface $\mathcal{S}$. We treat a surface as a differentiable manifold and build local charts (parametrizations) at all the estimated surface points. Thus, at $P \in \mathcal{S}$ we assume that the local chart (ψ, U):

$$\psi : U \subset \mathcal{S} \rightarrow \mathbb{R}^2$$

(ψ a diffeomorphism, $P \in$ open set U) is such that $\psi(P) = (0, 0)$ and its imbedding:

$$\phi = i \circ \psi^{-1} : \psi(U) \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

in $\mathbb{R}^3$ (based on P's tangent plane coordinates) is the graph of some function h: $\psi(U) \rightarrow \mathbb{R}$ with:

$$h(0, 0) = 0$$

$$h_p(0, 0) = \left.\frac{\partial h}{\partial p}\right|_{(0,0)} = 0$$

$$h_q(0, 0) = \left.\frac{\partial h}{\partial q}\right|_{(0,0)} = 0$$

(this is always true in some local chart). The Taylor expansion of h about the origin is:

$$h(p, q) = \frac{1}{2}(h_{pp}\, p^2 + 2h_{pq}\, pq + h_{qq}\, q^2) + R$$

Since our ultimate goal is computation of curvatures and related information, we take the simplest local chart which is appropriate, i.e. where:

$$h = \frac{1}{2} ep^2 + fpq + \frac{1}{2} gq^2 \qquad (1)$$

where we write:

$$e = h_{pp}(0, 0), \quad f = h_{pq}(0, 0), \quad g = h_{qq}(0, 0)$$

(known as a *parabolic quadric*).

Surface curvature

The curvature of the surface $\mathcal{S}$ at P can be computed from its local parametrization $\phi : \psi(U) \rightarrow \mathbb{R}^3$ in P's tangent plane coordinates. The surface normal at P is expressed as:

$$\mathbf{N}(0, 0) = \left. \frac{\phi_p \times \phi_q}{\| \phi_p \times \phi_q \|} \right|_{(0, 0)}$$

(In the following, we take it as understood that derivatives are evaluated at (0, 0), i.e. the $|_{(0,0)}$ is implicit.) The matrices

$$F_1 = \begin{pmatrix} (\phi_p, \phi_p) & (\phi_p, \phi_q) \\ (\phi_q, \phi_p) & (\phi_q, \phi_q) \end{pmatrix}$$

$$F_2 = \begin{pmatrix} -(\phi_p, \mathbf{N}_p) & -(\phi_p, \mathbf{N}_p) \\ -(\phi_q, \mathbf{N}_p) & -(\phi_q, \mathbf{N}_q) \end{pmatrix}$$

are determined from the first and second fundamental forms respectively of the surface ($(\bullet, \bullet)$ denotes inner product). The principal curvatures κ_1, κ_2 of ϕ at P in $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$ coordinates are the two eigenvalues of the matrix $F_2 F_1^{-1}$, and the Gaussian and mean curvatures are:

$$\kappa_g = \kappa_1 \kappa_2,$$

$$\kappa_m = \frac{\kappa_1 + \kappa_2}{2}$$

respectively. We show how to compute uncertainties in the curvatures below.

RECURSIVE ESTIMATION OF SURFACE PARAMETERS

Instantiating the model

Now, we wish to determine the local quadric surface passing through point P which 'best' (in a sense made precise below) fits neighbouring points $Q_i = (p_i, q_i, n_i)^t$ and their normals $\mathbf{n}_i = (\alpha_i', \beta_i', \gamma_i')^t$. In P's tangent plane coordinate system, the equation of the quadric gives us a first measurement equation:

$$E_1(e, f, g) = p_i^2 e + 2 p_i q_i f + q_i^2 g - 2 n_i = 0 \qquad (2)$$

between the position of Q_i and the parameters e, f, g which are to be determined from the data.

In addition to estimated surface point locations, the 3D gradient operator provides an estimate of gradient direction, and we can use the measured normal $\mathbf{n}_i$ at point Q_i to further constrain the quadric surface parameters. We know that, in the tangent plane coordinates, the quadric's normal at point Q_i is:

$$\mathbf{n}_i = \begin{pmatrix} -p_i e - q_i f \\ -p_i f - q_i g \\ 1 \end{pmatrix}$$

Denoting the scaled normal *measured* at point Q_i by $(\alpha_i, \beta_i, 1)^t = (\alpha_i'/\gamma_i', \beta_i'/\gamma_i', \gamma_i'/\gamma_i')^t$, we obtain two more measurement equations:

$$E_2(e, f, g) = p_i e + q_i f + \alpha_i = 0 \qquad (3)$$

$$E_3(e, f, g) = p_i f + q_i g + \beta_i = 0 \qquad (4)$$

Equations $(E_1 - E_3)$ are the three measurement equations which constrain the determination of the parameters e, f, g of the quadric at point P. These equations should be compared to the four equations $E_1' - E_4'$ of Sander and Zucker[10]. Equation E_1 is the same, but equations $(E_2' - E_4')$ there were based on unit normals and involved a non-linear combination of e, f, g. The only restriction on our equations here is the assumption that the normal $\mathbf{n}_i$ of Q_i measured in the tangent plane coordinates of P has a nonzero third component, which is reasonable if we assume that Q_i lies in the neighbourhood of point P. (In fact, if the surface is regular, such a neighbourhood exists[15] at least before discretization*). When this component vanishes, the local parametrization of the quadric in these coordinates is no longer valid, and point Q_i should not be taken into account.

Denoting:

$$\mathbf{A}_i = \begin{pmatrix} p_i^2 & 2 p_i q_i & q_i^2 \\ p_i & q_i & 0 \\ 0 & p_i & q_i \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} 2 n_i \\ -\alpha_i \\ -\beta_i \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} e \\ f \\ g \end{pmatrix}$$

the measurement equations $(E_1 - E_3)$ at P can be put in matrix form:

$$\mathbf{A}_i \mathbf{x} = \mathbf{b}_i$$

Non-recursive minimum variance least-squares solution

We wish to weight the measurement equations by the uncertainty of our measured parameters, i.e. the coordinates of points Q_i and attached normals $\mathbf{n}_i$. Once this is done (*cf.* next sections), we end up with a matrix $\mathbf{W}_i$ which is the covariance of $\mathbf{A}_i \mathbf{x} - \mathbf{b}_i$:

$$\mathbf{W}_i = E[(\mathbf{A}_i \mathbf{x} - \mathbf{b}_i)(\mathbf{A}_i \mathbf{x} - \mathbf{b}_i)^t]$$

Then a weighted least-squares solution $\mathbf{x}$ to our problem at P using all Q_i, $i = 1, \ldots, n$ in some neighbourhood will therefore minimize:

$$C = \sum_i (\mathbf{A}_i \mathbf{x} - \mathbf{b}_i)^t \mathbf{W}_i^{-1} (\mathbf{A}_i \mathbf{x} - \mathbf{b}_i)$$

and is given by:

$$\mathbf{x} = (\mathbf{A}^t \mathbf{W}^{-1} \mathbf{A})^{-1} \mathbf{A}^t \mathbf{W}^{-1} \mathbf{b}$$

*A common enough assumption throughout computer vision.

where:

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{A}_n \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{b}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{b}_n \end{pmatrix} \quad \text{and} \quad \mathbf{W} = \begin{pmatrix} \mathbf{W}_1 & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \mathbf{W}_n \end{pmatrix}$$

Recursive solution

In fact, we implement a recursive solution to this problem, better known as a *Kalman filter*[16,17]. By this method, each time a new measurement Q_i is given, it is only necessary to compute $\mathbf{A}_i$ and $\mathbf{b}_i$ *for that point* and to update the current solution $(\mathbf{x}_i, \mathbf{S}_i)$ using the recursive equations:

$$\begin{cases} \mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{K}_i(\mathbf{b}_i - \mathbf{A}_i \mathbf{x}_{i-1}) \\ \mathbf{K}_i = \mathbf{S}_{i-1} \mathbf{A}_i^t (\mathbf{W}_i + \mathbf{A}_i \mathbf{S}_{i-1} \mathbf{A}_i^t)^{-1} \\ \mathbf{S}_i = (\mathbf{I} - \mathbf{K}_i \mathbf{A}_i)\, \mathbf{S}_{i-1} \end{cases}$$

$\mathbf{S}_i$ is the *parameter* covariance matrix:

$$\mathbf{S}_i = E[(\mathbf{x}_i - \mathbf{x})\ (\mathbf{x}_i - \mathbf{x})^t]$$

relating the current estimate $\mathbf{x}_i$ and the ideal value $\mathbf{x}$ of the parameter vector. This is a measure of the quality of our estimate – a small covariance means that the computed estimate $\mathbf{x}_i$ is expected to lie close to the 'actual' parameters $\mathbf{x}$. It is necessary to initialize the filter with $(\mathbf{x}_0, \mathbf{S}_0)$, which can be taken as $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{S}_0 \mathbf{x} \infty \mathbf{I}$ when *no a priori* information is available about any of the parameters.

Practical details

For simplicity, the preceding development was presented with all data assumed to be in P's tangent plane coordinate system. We now show how to transform from the actual data points Q_i and normal vectors $\mathbf{n}_i$, each with their respective associated covariance matrices $\mathbf{W}_Q$ and $\mathbf{W}_\mathbf{n}$, measured in a *global* coordinate system, i.e. the coordinate system of the image, to variables $\mathbf{v} = (p_i, q_i, n_i, \alpha_i, \beta_i)^t$ and associated covariance matrix $\mathbf{W}_i$ in P's tangent plane coordinates. Thus we can apply the above theory directly to the image data.

Computing parameters v

We now assume that $Q_i = (x_i, y_i, z_i)^t$ and $\mathbf{n}_i = (n_{x_i}, n_{y_i}, n_{z_i})^t$ are given in a global coordinate system $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. To express them in P's tangent plane coordinates (P, Q, N) at point $P = (x, y, z)^t$, we compute:

$$\begin{pmatrix} p_i \\ q_i \\ n_i \end{pmatrix} = \mathbf{R} \begin{pmatrix} x_i - x \\ y_i - y \\ z_i - z \end{pmatrix}, \quad \begin{pmatrix} \alpha_i' \\ \beta_i' \\ \gamma_i' \end{pmatrix} = \mathbf{R} \begin{pmatrix} n_{x_i} \\ n_{y_i} \\ n_{z_i} \end{pmatrix}$$

and:

$$\alpha_i = \frac{\alpha_i'}{\gamma_i'}, \quad \beta_i = \frac{\beta_i'}{\gamma_i'}$$

for:

$$\mathbf{R} = \begin{pmatrix} P_x & P_y & P_z \\ Q_x & Q_y & Q_z \\ N_x & N_y & N_z \end{pmatrix}$$

The coordinates of the tangent plane basis vectors at P expressed in the global image coordinates $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ are:

$$\mathbf{P} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}, \quad \mathbf{Q} = \begin{pmatrix} Q_x \\ Q_y \\ Q_z \end{pmatrix}, \quad \mathbf{N} = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix}$$

Computing covariances $\mathbf{W}_i$

We assume that the covariance of point Q_i and its normal $\mathbf{n}_i$ are given in the *global coordinate system* $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ by $\mathbf{W}'_{Q_i}$ and $\mathbf{W}'_{\mathbf{n}_i}$, respectively. Since, for any affine transformation of a random variable $\mathbf{v}_i \rightarrow \mathbf{w}_i = \mathbf{M}(\mathbf{v}_i - \mathbf{v})$ we have:

$$E[(\mathbf{w}_i - \mathbf{w})\ (\mathbf{w}_i - \mathbf{w})^t] = \mathbf{M}\, E[(\mathbf{v}_i - \mathbf{v})\ (\mathbf{v}_i - \mathbf{v})^t]\, \mathbf{M}^t$$

the corresponding covariance matrices expressed in P's tangent plane coordinate system $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$ are:

$$\mathbf{W}_{Q_i} = \mathbf{R} \mathbf{W}'_{Q_i} \mathbf{R}^t, \quad \mathbf{W}_{\mathbf{n}_i} = \mathbf{R} \mathbf{W}'_{\mathbf{n}_i} \mathbf{R}^t$$

which are 3×3 matrices.

In fact, if we express $\mathbf{n}_i = (\alpha_i', \beta_i', \gamma_i')^t$ in the tangent plane coordinates, we must compute the covariance of $\alpha_i = \alpha_i'/\gamma_i'$ and $\beta_i = \beta_i'/\gamma_i'$. As a first order approximation, we compute the 2×2 matrix $\tilde{\mathbf{W}}_{\mathbf{n}_i}$:

$$\tilde{\mathbf{W}}_{\mathbf{n}_i} = \mathbf{J}_1 \mathbf{W}_{\mathbf{n}_i} \mathbf{J}_1^t$$

where $\mathbf{J}_1$ is the Jacobian matrix:

$$\mathbf{J}_1 = \begin{pmatrix} \dfrac{1}{\gamma_i'} & 0 & \dfrac{-\alpha_i'}{\gamma_i'^2} \\ 0 & \dfrac{1}{\gamma_i'} & \dfrac{-\beta_i'}{\gamma_i'^2} \end{pmatrix}$$

of the change of variables. Therefore the 5×5 matrix:

$$\mathbf{W}_{Q,\mathbf{n}} = \begin{pmatrix} \mathbf{W}_Q & \mathbf{0}_{3\times 2} \\ \mathbf{0}_{2\times 3} & \mathbf{W}_{\mathbf{n}_i} \end{pmatrix}$$

is the covariance of our measurement vector $(p_i, q_i, n_i, \alpha_i, \beta_i)$.

The 3×3 covariance matrix $\mathbf{W}_i$ is computed as a first order approximation by:

$$\mathbf{W}_i = \mathbf{J}_2 \mathbf{W}_{Q,\mathbf{n}_i} \mathbf{J}_2^t$$

where $\mathbf{J}_2$ is the Jacobian matrix:

$$\mathbf{J}_2 = \begin{pmatrix} 2p_i e + 2q_i f & 2p_i f + 2q_i g & -2 & 0 & 0 \\ e & f & 0 & 1 & 0 \\ 0 & f & g & 0 & 1 \end{pmatrix}$$

The 2×2 covariance matrix of curvatures is determined similarly:

$$W_c = J_5 W_Q J_5'$$

with the Jacobian:

$$J_5 = \begin{pmatrix} c & -2b & a \\ \frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix}$$

ESTIMATING ERROR IN EDGE DETECTION

In this section we determine the uncertainty inherent in edge detection in digital images by considering the 3D (modified Canny) edge detector of Monga and Deriche[3,4]. We first deal with uncertainty in edge position, and then with the error in edge direction and magnitude. We determine precisely the covariance matrices needed for the local quadric surface fitting described above, and derive the very interesting result that the uncertainty in edge position and magnitude is highly dependent on the orientation of the edge with respect to the image coordinate axes.

Error in edge position

The Monga-Deriche 3D modified Canny edge detector introduces two 1D linear filters:

$$S(x) = \frac{\alpha}{4}(\alpha|x| + 1)\, e^{-\alpha|x|} \quad \text{'smoothing' component}$$

$$L(x) = \alpha^2(1 - \alpha|x|)\, e^{-\alpha|x|} \quad \text{'Laplacian' component}$$

The image is convolved with the impulse response:

$$D(x, y, z) = L(x)S(y)S(z) + S(x)L(y)S(z) + S(x)S(y)L(z)$$

which is an approximation to the local image Laplacian,

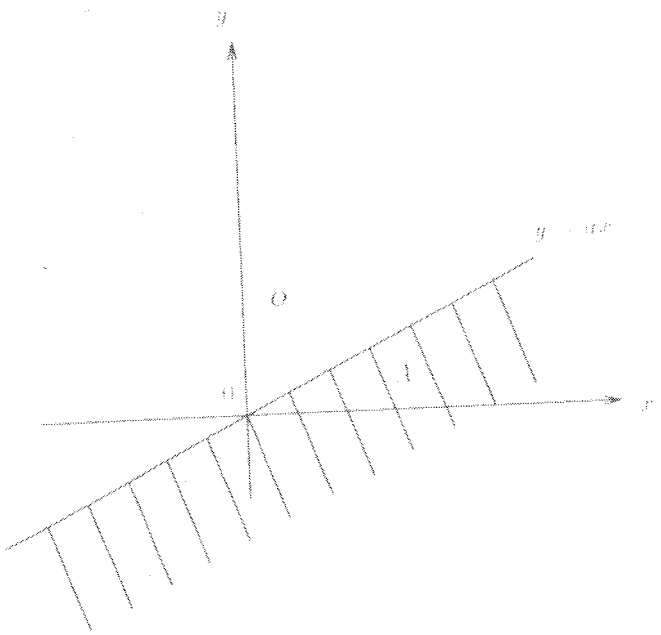

Figure 1. Ideal 2D edge

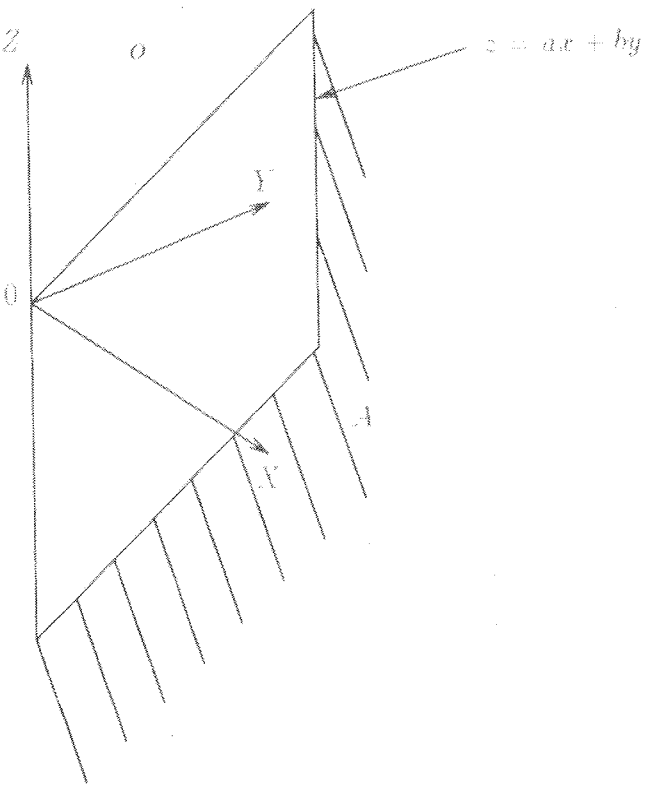

Figure 2. Ideal 3D edge

and edges are then detected as the zero-crossings of D.

The extensions of Canny's model in 2D and 3D are respectively a straight line and a plane dividing the space into two areas of constant grey level (A and O) (see Figures 1 and 2).

To compute the uncertainty in edge localization, we consider the 'perfect 3D edge' as being defined by a plane:

$$H(x, y, z) = \begin{cases} 0 & \text{if } z \geqslant ax + by \\ A & \text{if } z \leqslant ax + by \end{cases}$$

with $a, b \geqslant 0$. A noisy edge:

$$C(x, y, z) = H(x, y, z) + \eta(x, y, z)$$

is modelled by corrupting $H(x, y, z)$ with zero-mean additive Gaussian noise $\eta(x, y, z)$ (a generalization of the Canny 1D edge model).

We assume that the noise is separable into $\eta(x, y, z) = \eta_1(x) + \eta_2(y) + \eta_3(z)$ where the η_i's are independent zero-mean Gaussian noises, with covariance η_0^2. A point (x_0, y_0, z_0) is determined as being an edge point if:

$$\theta_0 = \theta(x_0, y_0, z_0) = \iiint C(x, y, z)\, D(x - x_0, y - y_0, z - z_0)\, dx\, dy\, dz = 0$$

i.e.:

$$\theta_0 = [C * D](x_0, y_0, z_0) = 0$$

This convolution product can be rewritten as:

$$\theta_0 = P_0 + N_0$$

where $P_0 = [H * D]_0$ (ideal edge component) and $N_0 = [\eta * D]_0$ (noise component). We first compute:

$$P_0 = A \iiint_{z \le ax+by} D(x_0 - x,\ y_0 - y,\ z_0 - z)\ dx\ dy\ dz$$

and then approximate the result using a first order Taylor expansion of P_0 around the origin. We obtain*:

$$P_0 \simeq A\,(ax_0 + by_0 - z_0)\ \alpha^2\ f(a,\ b)$$

where:

$$f(a,\ b) = \frac{(a^3b^3 + 3a^2b^3 + 3ab^3 + b^3 + 7a^2b^2 + 3a^3b^2 + 3ab^2 + 3a^3b + 3a^2b + a3)(a^2 + b^2 + 1)}{(a+b)^3\ (a+1)^3\ (b+1)^3}$$

Also, by using the fact that $\int_{-\infty}^{+\infty} L(x)\ dx = 0$ and $\int_{-\infty}^{+\infty} S(x)\ dx = 1$, we have:

$$N_0 = \iiint (\eta_1(x) + \eta_2(y) + \eta_3(z))\ D(x_0 - x,\ y_0 - y,\ z_0 - z)\ dx\ dy\ dz$$

$$= \int \eta_1(x)\ L(x_0 - x)\ dx + \int \eta_2(y)\ L(y_0 - y)\ dy + \int \eta_3(z)\ L(z_0 - z)\ dz$$

Since θ_0 at edge points, point $(x_0,\ y_0,\ z_0)$ is an edge point when $P_0 + N_0 = 0$. Hence, for each detected edge point, we have $E[P_0^2] = E[N_0^2]$. Now, a simple computation shows that:

$$E[N_0^2] = \frac{3}{2}\ \alpha^3 \eta_0^2$$

$$E[P_0^2] = E[(ax_0 + by_0 - z_0)^2]\ A^2 \alpha^4 f^2(a,\ b)$$

giving the standard deviation of the edge position as:

$$\sigma\left[\frac{ax_0 + by_0 - z_0}{(a^2 + b^2 + 1)^{1/2}}\right] = \sqrt{\frac{3}{2}}\ \frac{\eta_0}{\alpha A}\ h(a,\ b) \qquad (5)$$

where:

$$h(a,\ b) = \frac{1}{f(a,\ b)\ \sqrt{a^2 + b^2 + 1}}$$

Equation (5) provides a quantitative estimate of the quality of edge localization. As in the 1D case, this shows that a large value of α yields good localization. The function $h(a,\ b)$ appears as a correction factor which depends on edge orientation, and which is minimum when the plane is parallel to one of the coordinates axes, i.e. when a and b both tend to zero or when one of them tends to infinity. In these three cases, $h(a,\ b) \to 1$. The maximum value of $h(a,b)$ is $h(1,\ 1) = 128/21\sqrt{3} \simeq 3.52$, obtained with edge planes whose normal is parallel to vector, $(1,\ 1,\ -1)$. In this case, the localization is more than 3.5 times less accurate than in the previous cases! This clearly demonstrates the importance of accounting for the uncertainty in edge localization *as a function of edge spatial orientation*.

*The computation was performed using the symbolic mathematics program Maple[18]. The listing of the result before evaluation of the symbolic expression is too long (30 pages) to be included in this publication.

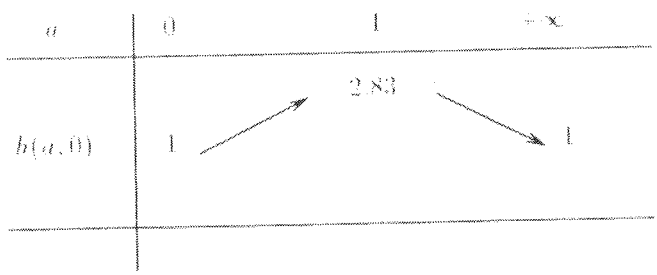

Figure 3. Change of monotony of h(a,0)

The 2D case is covered by setting $b = 0$, which yields:

$$h(a,\ 0) = \frac{1}{f(a,\ b)\ \sqrt{a^2 + 1}} = \frac{(a+1)^3}{(a^2+1)^{3/2}}$$

In this case, the minimum value of $h(a,\ 0)$ is 1 for $a = 0$ or $a \to \infty$ (edges parallel to the coordinate axes). The maximum value (producing the worst localization) is $2\sqrt{2} \simeq 2.83$ and is again obtained for $a = 1$ (see Figure 3).

Uncertainty of gradient magnitude

As we showed above, it is important to estimate the value of A to compute the accuracy of edge localization and to estimate the uncertainty in the edge direction. Information on both is provided by determining the edge gradient, which is computed by convolving the image with three derivative filters:

$$g_1(x,\ y,\ z) = d(x)\ S(y)\ S(z)$$

$$g_2(x,\ y,\ z) = S(x)\ d(y)\ S(z)$$

$$g_3(x,\ y,\ z) = S(x)\ S(y)\ d(z)$$

where $S(x)$ is the previously defined smoothing filter, and $d(x)$ is the derivative filter:

$$d(x) = cx\ e^{-\alpha|x|}$$

where c is a constant. For the ideal edge H:

$$G_{x_0} = G_x(x_0,\ y_0,\ z_0) = [H * g_1]_0, = \iiint H(x,\ y,\ z)\ g_1(x_0 - x,\ y_0 - y,\ z_0 - z\ dx\ dy\ dz$$

$$G_{y_0} = [H * g_2]_0$$

$$G_{z_0} = [H * g_3]_0$$

give the gradient vector $\mathbf{G}_0 = (G_{x_0},\ G_{y_0},\ G_{z_0})^t$, whose direction is orthogonal to the contour, i.e. $\mathbf{G}_0 \times (a,\ b-1)^t = 0$. However, the gradient magnitude thus obtained again depends on the orientation of the contour, and is given by†:

$$\|\mathbf{G}_0\| = Ag(a,\ b)$$

†For a point lying on the contour, i.e. $z_0 = ax_0 + by_0$.

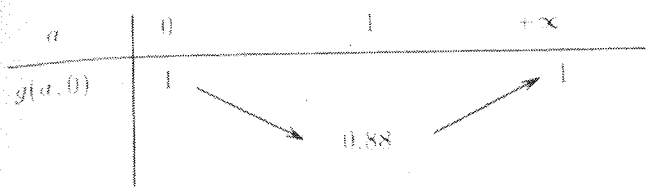

Figure 4. Change of monotony of g(a,0)

where $g(a, b)$ is the ratio of two polynomials in a and b:

$$g(a, b) = \frac{\begin{array}{r}(ab+b+a)(b^4(a+1)^2+ \\ b^3(3a^3+8a^2+8a+3)+b^2(a^4+8a^3+ \\ 15a^2+8a+1)+b(2a^4+8a^3+8a^2+ \\ 2a)+a^4+3a^3+a^2)\sqrt{a^2+b^2+1}\end{array}}{(b+1)^3\,(a+b)^3\,(a+1)^3}$$

It is interesting to study how $g(a, b)$ varies with direction (a, b). It has a maximum value of 1 for a contour plane parallel to one of the coordinate system axes, i.e. when a and b both tend to zero or when one of them tends to infinity, and takes its minimum value 0.85 for $(a, b) = (1, 1)$.

If we set $b = 0$, we have a 2D contour, and in this case:

$$g(a, 0) = \frac{(a^2+3a+1)\sqrt{a^2+1}}{(a+1)^3}$$

Again, the maximum value of 1 is obtained for $a = 0$ or $a \to \infty$. The minimum value $g(1, 0) = 0.88$ is obtained for $a = 1$ (see Figure 4).

Practical computation of uncertainty

Position uncertainty

It is now possible to include the correct values of a, b and A in equation (5) of the standard deviation of edge localization. This is done by computing first the gradient vector $\mathbf{G} = (G_x, G_y, G_z)$, giving:

$$(a, b) = (G_x/G_z, G_y/G_z)$$

We then compute the gradient magnitude $\|\mathbf{G}_0\|$, and divide it by the correcting factor $g(a, b)$ to obtain $A = \|\mathbf{G}_0\|/g(a, b)$. Putting these values back into equation (5), we obtain the standard deviation of edge localization:

$$\sigma_\perp = \sqrt{\frac{3}{2\alpha}}\,\frac{\eta_0}{\|\mathbf{G}_0\|}\,q(a, b)$$

where:

$$q(a, b) = g(a, b)\,h(a, b)$$

This covariance is finally included in a covariance matrix expressed in a coordinate system (**I**, **J**, **K**) *attached to the edge*, i.e. such that the **K** axis is parallel to $(a, b, -1)^t$, and the **I** and **J** axes are parallel to the edge plane. This covariance matrix has the form:

$$\Sigma_\perp = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_\perp^2 \end{pmatrix}$$

The important fact here is that the covariance $\sigma_\perp^2$ ranges between a minimum value of σ_{min} and a maximum value σ_{max} as a function on the 3D orientation of the edge, and that the ratio $\sigma_{min}/\sigma_{max}$ is close to 9.6!

Knowing the orientation of the **I**, **J** and **K** axes, we can eventually express $\Sigma_\perp$ in the (**X**, **Y**, **Z**) coordinate system of the image as

$$\Sigma_1 = \mathbf{R}_1\,\Sigma_\perp\,\mathbf{R}_1^t$$

where $\mathbf{R}_1$ is as above. Finally, digitization noise is taken into account by computing:

$$\Sigma_2 = \Sigma_\perp + \begin{pmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{pmatrix}$$

where σ_x, σ_y and σ_z are directly proportional to voxel size in the **I**, **J** and **K** directions, respectively.

Normal uncertainty

For the computation of the uncertainty of the surface normal direction, i.e. of its coordinates, recall that each component of $\mathbf{G}_0$ is obtained by convolving H with $g_i(x, y, z)$ for $i = 1,2$ or 3. The covariance of each component of $\mathbf{G}_0$ is:

$$\mathbf{W}_{G_x} = \mathbf{W}_{G_y} = \mathbf{W}_{G_z} = \frac{3}{2}\,\alpha\eta_0^2$$

Now, we divide $\mathbf{G}_0$ by $g_2(a, b)$ to compute a correctly scaled gradient vector, and this vector is used as the surface normal estimate denoted $\mathbf{n}_i$ above. Therefore, the covariance attached to this normal vector is:

$$\mathbf{W}_{\mathbf{n}_i} = \frac{3\alpha\eta_0^2}{2g^2(a, b)}\,\mathbf{I}_3$$

This is the covariance that was used in the experiments. Note that the computation applies to a point located exactly on the ideal edge. In fact, since we have estimated the uncertainty on edge localization, it should be possible to bound the error in position, and also to compute an upper bound on $\mathbf{W}_{\mathbf{n}_i}$. This will be done in the near future.

What about other edge detection operators?

The same computation scheme could be used for any linear edge detection operator. Notice that the behaviour of isotropic operators such as Gaussian filter and its derivative does not depend on the edge orientation. For Canny's operator (first derivative of a Gaussian) the uncertainty in the 1D, 2D and 3D cases are the same up to a multiplicative constant due to the noise. If we compare for all edge orientations the Gaussian and the Deriche filters, we obtain equivalent performances (the Deriche filter is better for an edge orientation close to X, Y, Z axis, and Gaussian is better for an edge orientation along diagonals). We choose the Deriche filter because its recursive implementation yields a substantial saving in computing time. This is

particularly important for 3D medical images where the huge amount of data makes the algorithmic complexity and the storage requirements key points for edge detection.

FROM CURVATURES TO TYPICAL FEATURES

For each edge point, the previous section determines the Gaussian and mean curvatures, principal curvature directions, and the corresponding covariance matrices. Note that the scale is defined by the size of the neighbourhood used to fit the local geometric model. In this section we deal with the extraction of more global curvature features from the local curvature information.

Local curvature maps

A practical way of characterizing the behaviour of the curvature in the neighbourhood of a point is to define local curvature maps (technically the pullback of the field onto the tangent plane as used by Sander and Zucker[20] for the computation of the direction field index). Given a point P and its tangent plane defined by $(\mathbf{P}, \mathbf{Q}, \mathbf{N})$, let V be the intersection of a sphere whose centre is P and radius r with the set of the edge points (this defines a neighbourhood of P), and let W be the orthogonal projection of the points of V onto the tangent plane. At each point of W we attach the curvatures of its corresponding points in V. The size of V could be determined using the distance of the points to the tangent plane and the angle between the gradient at a point and the gradient at point P. We thus define a map characterizing the behaviour of the curvatures around P.

Extracting lines of curvature extrema

From the local curvature images we can extract, for instance, the maxima of the maximum curvature in the maximum curvature direction. This may be done similarly to the classical extraction of the extrema of the gradient magnitude in the gradient direction in 2D edge detection methods[13,14].

Let $C(P)$ be the local maximum curvature map attached to point P, and let $G(P)$ be the maximum curvature direction. We compare the value of the maximum curvature along the straight line defined by P and $G(P)$ and retain P if its maximum curvature is a local extrenum along this direction. Thus we obtain local curvature extrema candidates. To remove false extrema, we perform a 3D hysteresis thresholding using the mean curvature already determined. This is done analogously to the thresholding of the local gradient extrema in the edge detection scheme described in Monga *et al.*[4].

RESULTS

Extraction of 3D edges

We performed three dimensional edge extraction, with the methods presented above, and then hysteresis thresholding, using first the original gradient magnitude $\mathbf{G}_0$, and then the correctly scaled gradient magnitude $\mathbf{G}_0/g(a, b)$. Figure 5 shows a cross section of a 3D medical Magnetic Resonance Image of a body displayed at the level of the heart. Figure 6 shows the result of hysteresis thresholding on the original gradient magnitude, while Figure 7 shows the results obtained with the

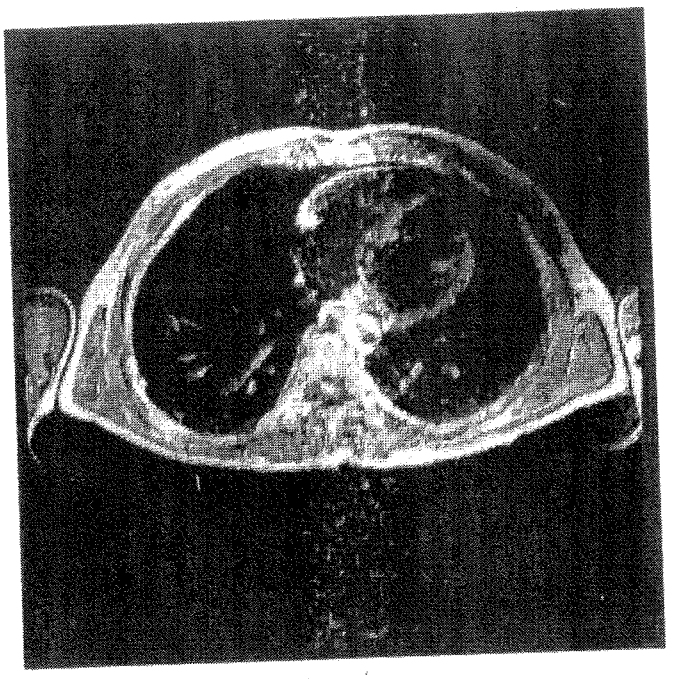

Figure 5. Original MR image

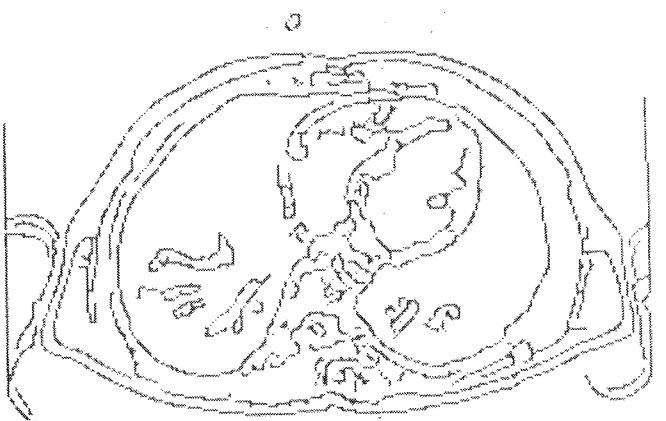

Figure 6. Hysteresis thresholding of edges with original gradient magnitude

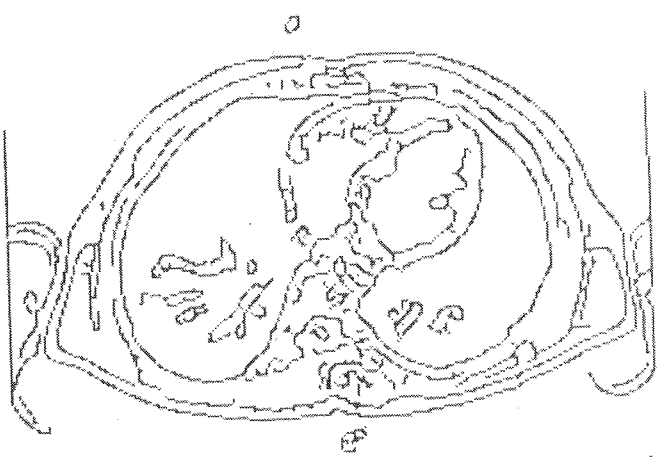

Figure 7. Hysteresis thresholding of edges with corrected gradient magnitude

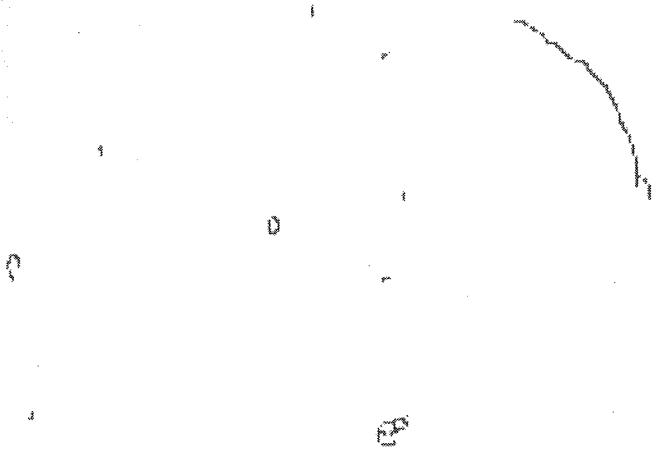

Figure 8. Difference between previous edge images

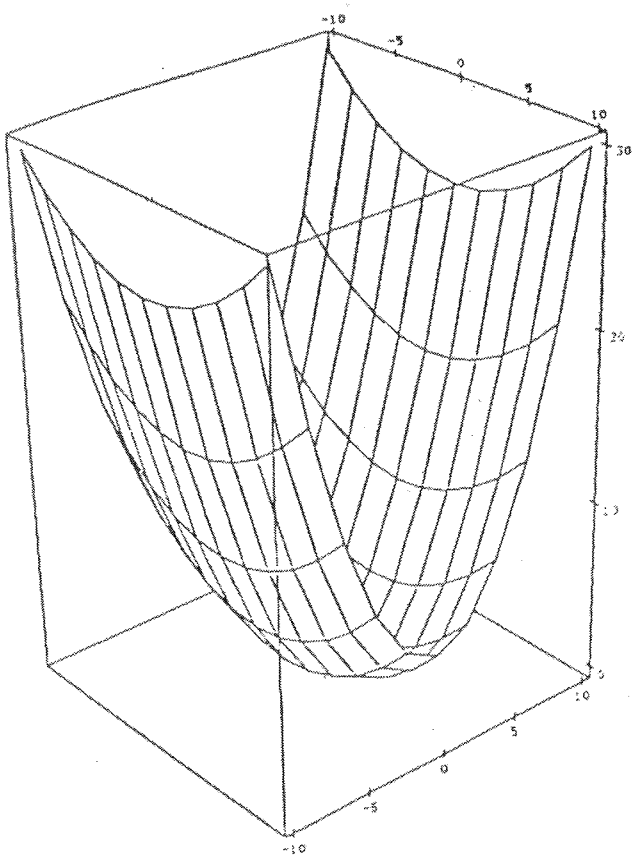

Figure 9. Elliptic paraboloid showing surface shape of the synthetic volume

correctly scaled gradient magnitude. The low and high thresholds are exactly the same for both image. Figure 10 shows the difference between the two results, illustrating the advantage of using the corrected value of gradient magnitude. We see that an important piece of contour is ignored if we do not correct the gradient magnitude.

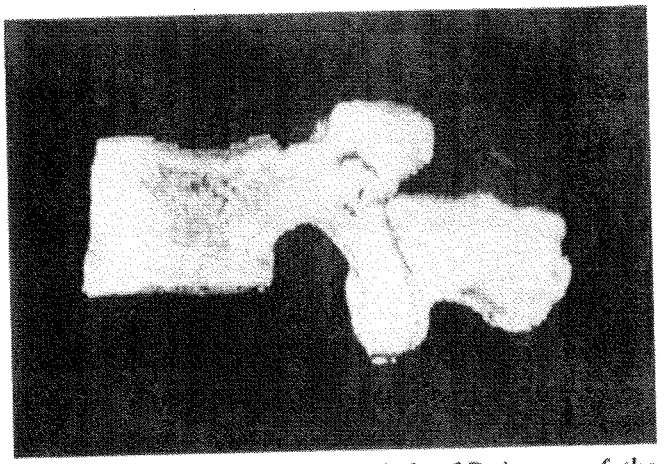

Figure 10. Perspective view of the 3D image of the vertebra for position A

Curvatures for synthetic data

We buit a synthetic volume whose implicit equation is given by:

$$z \leqslant ax^2 + by^2 + cx + xy + e$$

with $a = 1/20$, $b = 0.25$, $c = -4$, $d = -12$, $e = 224$ (see Figure 9 for a surface representation). This was done by creating a 3D digital image with points of maximum (resp. minimum) intensity within (resp. outside of) the volume. We extracted 3D edges with the algorithm described above. These edge points correspond to the surface of an ideal elliptic paraboloid.

First, note that there exists a single surface point T such that the equation of the *entire* surface takes the reduced form:

$$z' = \frac{1}{2} ex'^2 + fx'y' + \frac{1}{2} gy'^2$$

when (x', y', z') are expressed in the local tangent plane coordinate system attached to point T. This point is the vertex of the paraboloid and its coordinates are given by:

$$T = \left(\frac{-c}{2a}, \frac{-d}{2b}, \frac{-c^2}{4a} - \frac{d^2}{4b} + e \right)$$

For this particular point, the local quadric approximation is a *global* one, and the ideal parameters are $e = 2a$, $f = 0$, $g = 2b$. It is therefore possible to estimate these parameters with all the detected surface points (about 350 points). In this case the convergence is excellent towards the exact values. We show in Table 1 the results obtained with a smaller but still rather large number of edge points (86). We show successively the

Table 1. Estimation of the parameters (e, f, g) of the local quadrics

Point coordinates	*Ideal*	*Positions*	*+ Normals*	*+ Uncertainty*	*Predicted σ*
	$e = 0.1$	−0.0037	0.0172	0.102	0.011
	$f = 0$	0.0000	0.000	−0.003	0.008
(40,24,1)	$g = 0.5$	0.324	0.360	0.494	0.02
86 Neighbours					

Table 2. Estimation of gaussian and mean curvatures C_g and C_m

Point coordinates	*Ideal*	*Positions*	*+ Normals*	*+ Uncertainty*	*Predicted σ*
(40, 24, 1)	$C_g = 0.05$	−0.0012	0.0062	0.0505	0.014
86 Neighbours	$C_m = 0.3$	0.16	0.16	0.298	0.023
(47, 24, 2)	$C_g = 0.022$	0.089	0.063	0.0157	0.008
46 Neighbours	$C_m = 0.23$	0.333	0.297	0.2024	0.030
(45, 26, 3)	$C_g = 0.010$	−0.001	−0.0003	0.018	0.009
54 Neighbours	$C_m = 0.12$	0.053	0.107	0.143	0.018
(36, 22, 2)	$C_g = 0.010$	0.013	0.018	0.012	0.006
50 Neighbours	$C_m = 0.122$	0.133	0.172	0.127	0.015
(39, 23, 1)	$C_g = 0.031$	0.011	0.023	0.017	0.023
28 Neighbours	$C_m = 0.22$	0.10	0.25	0.21	0.062
(38, 28, 5)	$G_g = 0.004$	0.009	0.010	0.006	0.003
50 Neighbours	$C_m = 0.07$	0.105	0.11	0.09	0.01

results obtained for a least squares estimation with points only (measurement equation E1), then adding normals (measurement equations E2-E3). Finally, we show the results obtained when uncertainty on positions and normals is taken into account, following the computations of the previous sections. It is easy to check that not only is the estimate obtained much more accurate, but also that the computed standard deviation σ on the error estimation is perfectly coherent with the observed error.

At other points P on the surface, we applied our local quadric approximation with smaller neighbourhoods (containing about 50 points). The size of the neighbourhood is both controlled by limiting the angle between the neighbours' normal and P's normal, and also the distance between neighbours and P. We used the local approximation to compute locally the Gaussian and mean curvatures (C_g and C_m). In Table 2 we show the results obtained for this computation with the previous three methods. Here again, the results are much more accurate with the last approach, and the output covariance agrees almost perfectly with the observed errors.

Curvatures for real data

The goal of the experiments reported here is to show the stability of the mean and Gaussian curvature provided by our algorithm. We consider two 3D scanner images of the same vertebra obtained at two different positions A and B (see Figures 9 and 10). We perform the sequence of processes (edge detection, approximation of the curvatures) for a part of the verbetra. We obtain Gaussian and mean curvature images corresponding to positions A and B (see Figure 11, and Plates 1 and 2 on p. 416). We compute the rigid transformation (rotation and translation) from B to A : B-A using an algorithm described elsewhere[19]. We compare mean and Gaussian curvature images of vertebra A to the mean and Gaussian curvature images of vertebra B transformed by B-A (see Figure 11, and Plates 1 and 2 on p. 416). This allows a valid comparison of the curvatures computed for positions A and B. The stability of the results obtained for the two positions shows clearly the robustness of our algorithms.

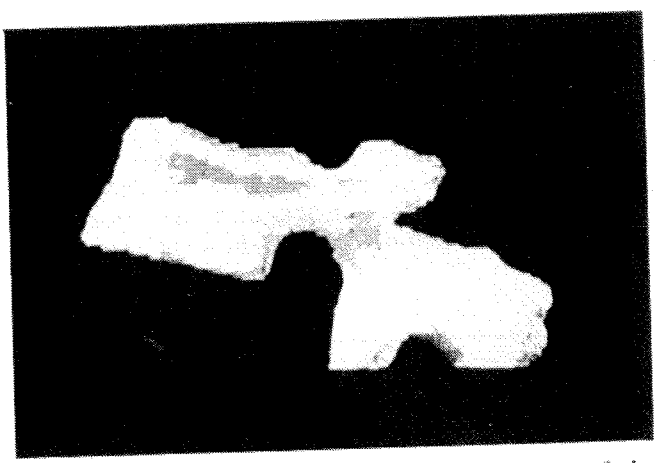

Figure 11. Perspective view of the 3D image of the vertebra for position B

Typical curvature features for synthetic and real data

Figures 12–20 and Plates 3–6 (p. 416–417) present some results for the determination of the extrema of the maximum curvature in the maximum curvature direction. We notice that our local approximation scheme provides a continuous maximum curvature field, allowing us to detect reliably and accurately the curvature extrema. Regarding our experiences, this continuity is mainly due to the continuity of the orientation of the gradient used for the local approximation.

Using covariance matrices in our least mean square criterion introduces a ponderation between the equations taking into account the position of the points and the normal orientations. For the original data corresponding to Figures 12–18 and Plates 4–6 (p. 416–417) the step edges have a very strong amplitude. This implies that the localization criterion is overestimated due to the first order approximation (the localization is inversely proportional to the step amplitude), and therefore the weight put on the position equations is too high. We also remark that the gradient coordinates are real values, but that the point coordinates are integer values which could induce false discontinuity. Given that each point produces one measurement equation using its position and two measurement equations using the orientation of its gradient, we can evaluate the ratio between point

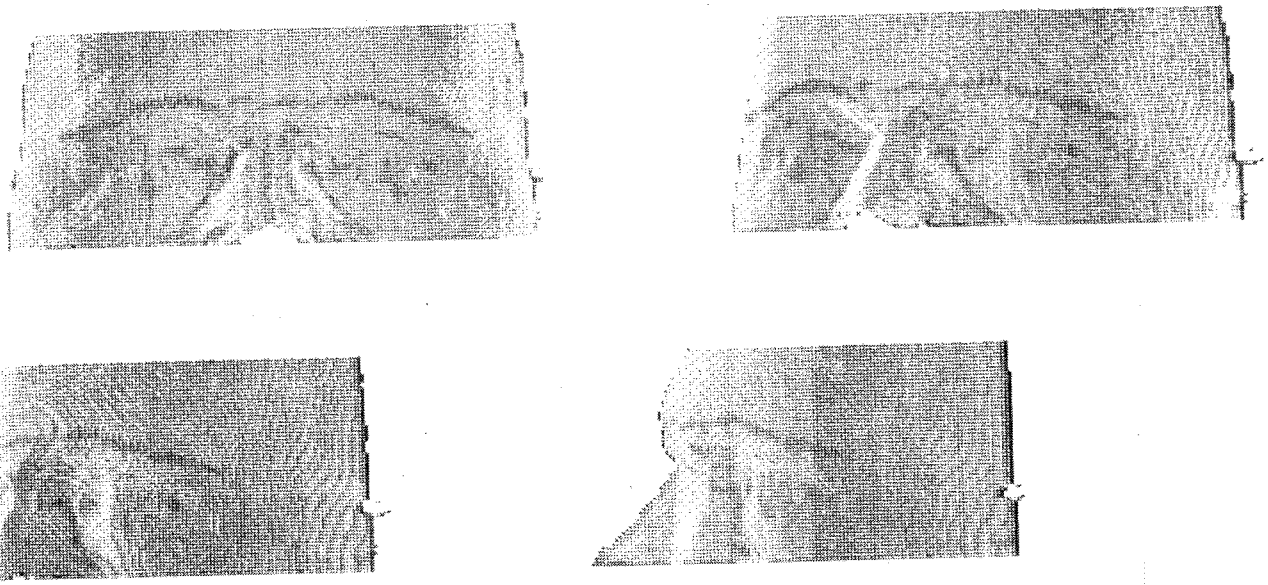

Figure 12. Perspective views of the 3D edges matched with the extrema of the maximum curvature in the maximum curvature direction coloured in dark (the ratio defining the weighting point/normal in the least mean squares is about 1/40)

Figure 13. Projection of the extrema of the maximum curvature in the maximum curvature direction corresponding to the previous figure (the ratio defining the weighting point/normal in the least mean squares is about 1/40)

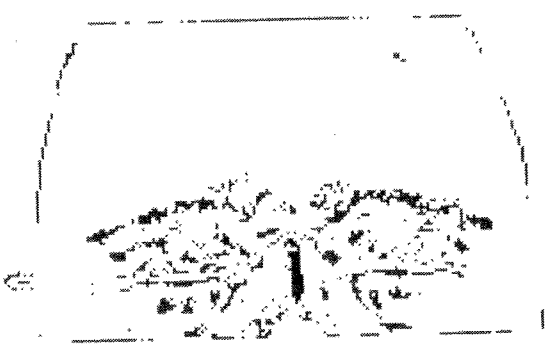

Figure 15. Projection of the extrema of the maximum curvature in the maximum curvature direction (the ratio defining the weighting point/normal in the least mean squares is about 1/320)

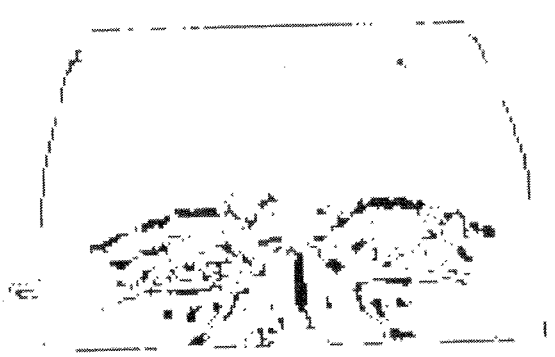

Figure 14. Projection of the extrema of the maximum curvature in the maximum curvature direction (the ratio defining the weighting point/normal in the least mean squares is about 1/40); the algorithm providing the extrema is slightly different

Figure 16. Projection of the extrema of the maximum curvature in the maximum curvature direction (the ratio defining the weighting point/normal in the least mean squares is about 1/5)

information and normal information. If we apply exactly the theoretical calculus presented before for these data, we obtain a ratio of 1/12 (1 for point and 12 for normal). This allows to obtain rather good results but where some false discontinuities still remain. Experimentally a ratio of 1/40 yields a good trade-off between the smoothness and the preservation of the singularities. The distortion of the theoretical optimum and the experimental one is due to the reasons we reported here. We also perform some experiences with a ratio of 1/5 and we obtain a bad continuity for the maximum curvature field.

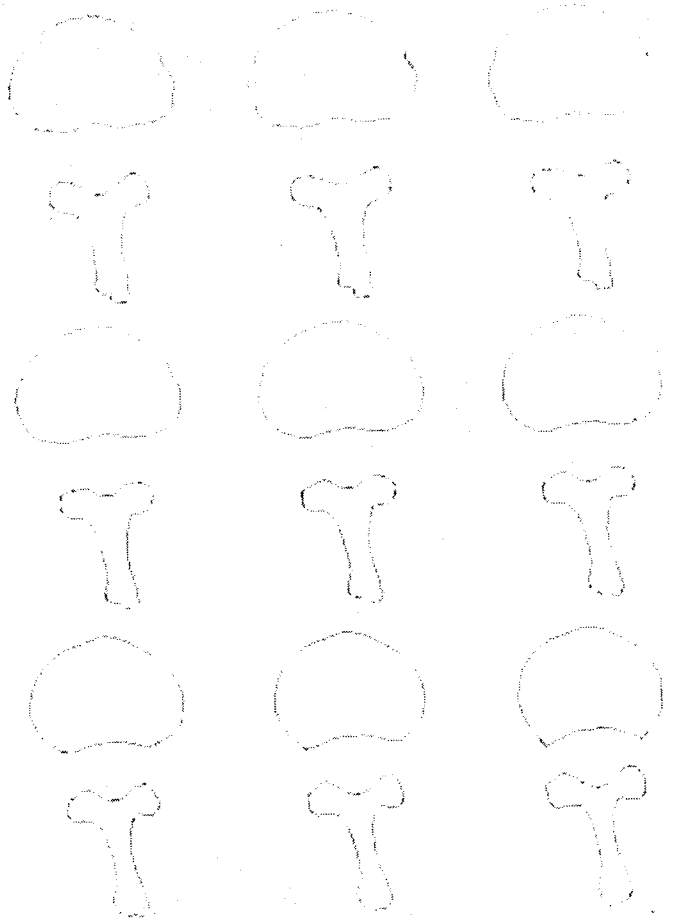

Figure 17. Cross sections of the vertebra for position A (see Figure 10) where the local curvature extrema are dark (left to right and top to bottom)

The main practical conclusion of our experiences is that the gradient orientation (approximating the orientation of the normal to the surface) is a strong regularization criteria for the local approximation. This illustrates the applicability of our theoretical developments, although its direct applicability is spoiled by first order approximations and by discretization.

Figure 19. Cross section of the vertebra for position B (see Figure 11) where the local curvature extrema are dark (left to right and top to bottom)

CONCLUSION

Our main objective was to develop robust and reliable tools useful for modelling and analysing surfaces of 3D objects. In this paper we showed the importance of a careful quantitative analysis of the various sources of uncertainty for computing second order derivative features (mean and Gaussian curvatures) on a discrete surface.

We analysed *quantitatively* the uncertainty in edge position, orientation and magnitude produced by the

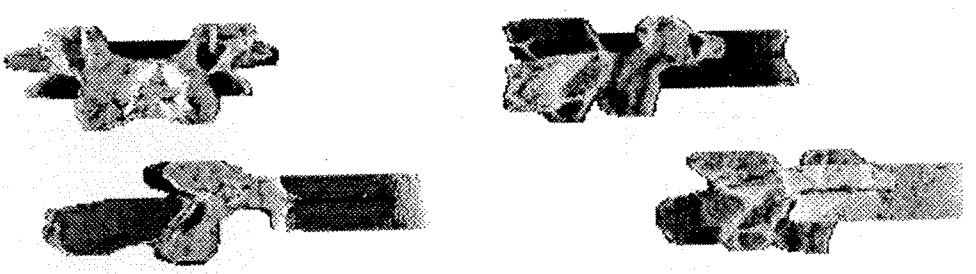

Figure 18. Perspective views corresponding to Figure 17

Figure 20. Perspective views corresponding to Figure 19

multidimensional (2D and 3D) versions of the Monga–Deriche–Canny recursive separable edge-detector. We showed, for instance, that depending on the orientation, the position uncertainty may vary with a ratio larger than 2.8 in the 2D case, and 3.5 in the 3D case. We showed that to estimate edge magnitude, the gradient magnitude must be corrected by a factor which depends on the orientation, and whose relative variation is close to 20%.

Then we revisited the algorithm initially proposed by Sander and Sucker for locally estimating the curvature of a discrete 3D surface, and we modified the original measurement equations and proposed an optimal estimation scheme to account for the previously computed uncertainties and corrections.

We tested the corrected edge detector on 2D and 3D medical images, and showed the importance of the corrected edge magnitude for edge detection. We also tested the surface modelling algorithm on discrete 3D objects – not only are the results obtained more accurate, but the computed measure of uncertainty attached to the results agrees extremely well with the true one.

We also show how to use these curvatures to determine typical curvature features on which registration and/or tracking procedures can robustly rely.

ACKNOWLEDGEMENTS

We thank Serge Benayoun for the implementation of the extraction of the local curvature extrema. Nathalie Gaudechoux provided a substantial help in the preparation of this manuscript.

We thank Pr. Bittoun of the Kremlin Bicetre Hospital in Paris and Pr. J. L. Coatrieux in Rennes which provided the medical images.

This work has been partially supported by Ge-Cgr and some of the presented images were acquired in collaboration with the Advanced Image Processing Group of GE-CGR in Buc, France. We shall give the details of this acquisition process, and a thorough description of the results very soon in a forthcoming joint paper Inria-Ge-Cgr.

This work was partially supported by Digital Equipment Corporation and European AIM (Advanced Informatics in Medicine) Project Murim.

REFERENCES

1 **Ayache, N, Boissonnat, J D, Cohen, L, Geiger, B, Levy-Vehel, J, Monga, O and Sander, P** 'Steps toward the automatic interpretation of 3d images', *Proc. NATO Advanced Research Workshop on 3D Imaging in Medicine*, Travemünde (June 1990)

2 **Zucker, S W and Hummel, R M** 'A three-dimensional edge operator', *IEEE Trans. PAMI*, Vol 3 No 3 (May 1981) pp 324–331

3 **Monga, O and Deriche, R** '3d edge detection using recursive filtering', *Conf. on Vision and Patt. Recogn.*, San Diego, CA (June 1989)

4 **Monga, O, Deriche, R and Rocchisani, J-M** '3d edge detection using recursive filtering: Application to scanner images', *Comput. Vision Graph. & Image Process.*, Vol 53 No 1 (January 1991) pp 76–87

5 **Brady, M, Ponce, J, Yuille, A and Asada, H** 'Describing surfaces', *Hideo Hanafusa and Hirochika Inoue* (eds), *Proc. Second Int. Symposium on Robotics Res.*, MIT Press Cambridge, MA (1985) pp 5–16

6 **Ponce, J and Brady, M** 'Toward a surface primal sketch', *Proc. IJCAI* (1985)

7 **Koenderink, J J** 'The internal representation of solid shape and visual exploration', in **Spillman and Wooten** (eds), *Sensory Experience, Adaptation, and Perception*, Lawrence Erlbaum, NJ (1984) pp 123–142

8 **Koenderink, J J** *Solid Shape*, MIT Press, Boston, MA (1990)

9 **Sander, P T and Zucker, S W** 'Tracing surfaces for surfacing traces', *Proc. 1st Int.Conf. on Comput. Vision*, London, UK (June 1987) pp 241–249

10 **Sander, P T and Zucker, S W** 'Inferring surface trace and differential structure from 3-D images', *IEEE Trans. PAMI*, Vol 12 No 9 (September 1990)

11 **Monga, O, Deriche, R, Malandain, G and Cocquez** 'Recursive filtering and edge closing: two primary tools for 3d edge detection', *Proc. 1st Euro. Conf. on Comput. Vision*, Springer-Verlag, Berlin (1990)

12 **Besl, P J and Jain, R C** 'Segmentation through Variable-Order surface fitting', *IEEE Trans. PAMI*, Vol 10 No 2 (March 1988) pp 167–192

13 **Deriche, R** 'Using Canny's criteria to derive a recursively implemented optimal edge detector', *Int. J. Comput. Vision* (1987) pp 167–187

14 **Canny, J** 'A computational approach to edge detection', *IEEE Trans. PAMI*, Vol 8 No 6 (November 1988) pp 679–698

15 **do Carmo, M P** *Differential Geometry of Curves and Surfaces*, Prentice-Hall, NJ (1976) pp 164

16 **Luenberger, D G** *Optimization by Vector Space Methods*, Wiley, New York (1969)

17 **Ayache, N** *Artificial Vision for Mobile Robots – Stereo-Vision and Multisensory Perception*, MIT Press, Boston, MA (1991)

18 **Char, B W, Geddes, O, Gonnet, G H and Watt, S W** *Maple Reference Manual, 4th edition*, Department of Computer Science, University of Waterloo, Ontario (1985)

19 **Bajcsy, R and Kovacic, S** 'Multiresolution elastic matching', *Comput. Vision, Graph., & Image Process.*, Vol 46 (1989) pp 1–21

20 **Sander, P T and Zucker, S W** 'Singularities of principal direction fields from 3-D images', *IEEE Trans. PAMI*, (to appear)

From voxel to intrinsic surface features

by O Monga *et al.*

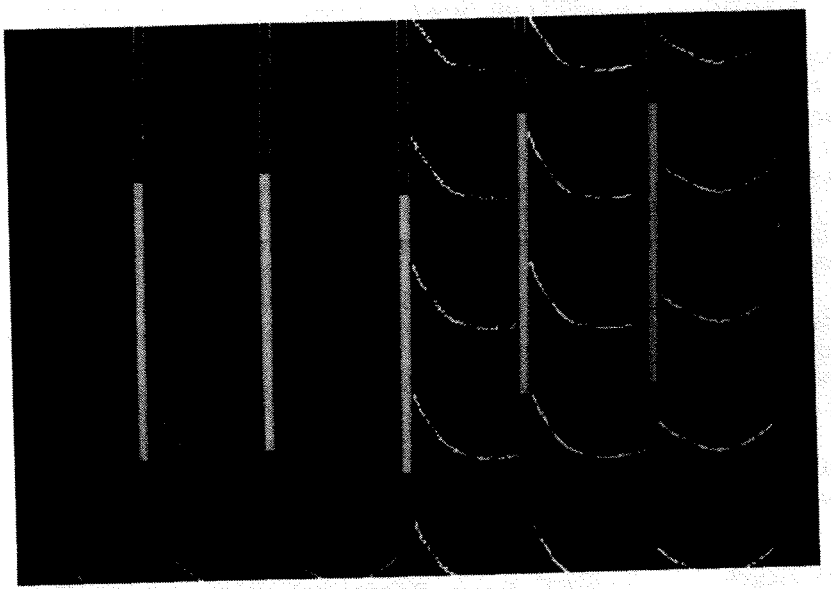

Plate 1. 1st column: 3D image (5 successive planes) corresponding to mean curvature for position A; 2nd column: 3D image corresponding to mean curvature for position B transformed by B-A; 3rd column: 3D image corresponding to mean curvature for position B; 4–6th columns: as previous columns, respectively, but for Gaussian curvatures. (Size of neighbourhood used for the local approximation for these results is 5)

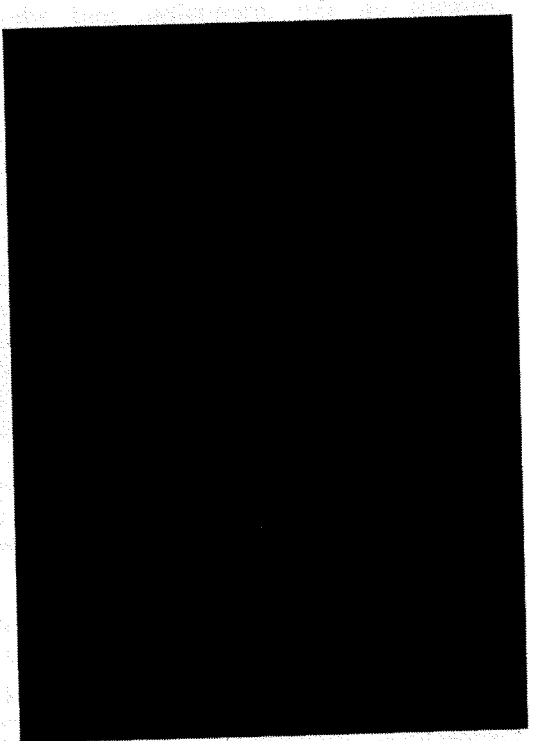

Plate 2. As columns 1 and 2 of Plate 1, but using a neighbourhood of size 10 for the local approximation

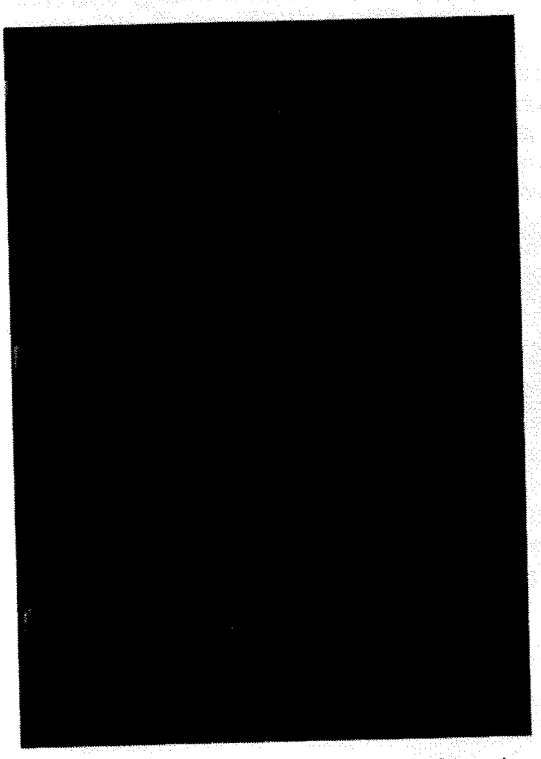

Plate 3. As Plate 1, but for Gaussian curvature

Plate 4. Sign of the Gaussian curvatue for a NMR image of the head where we have extracted the part corresponding to the face. Green = hyperbolic points; red = points where Gaussian curvature is zero; grey = elliptic points (the ratio defining the weighting point/normal in the least mean squares is about 1/40)

Plate 5. Maximum curvature map. Red = extremum of the maximum curvature in the maximum curvature direction (the ration defining the weighting point/normal in the least mean squares is about 1/40)

Plate 6. Extrema of the maximum curvature in the maximum curvature direction (in red) corresponding to Plate 5